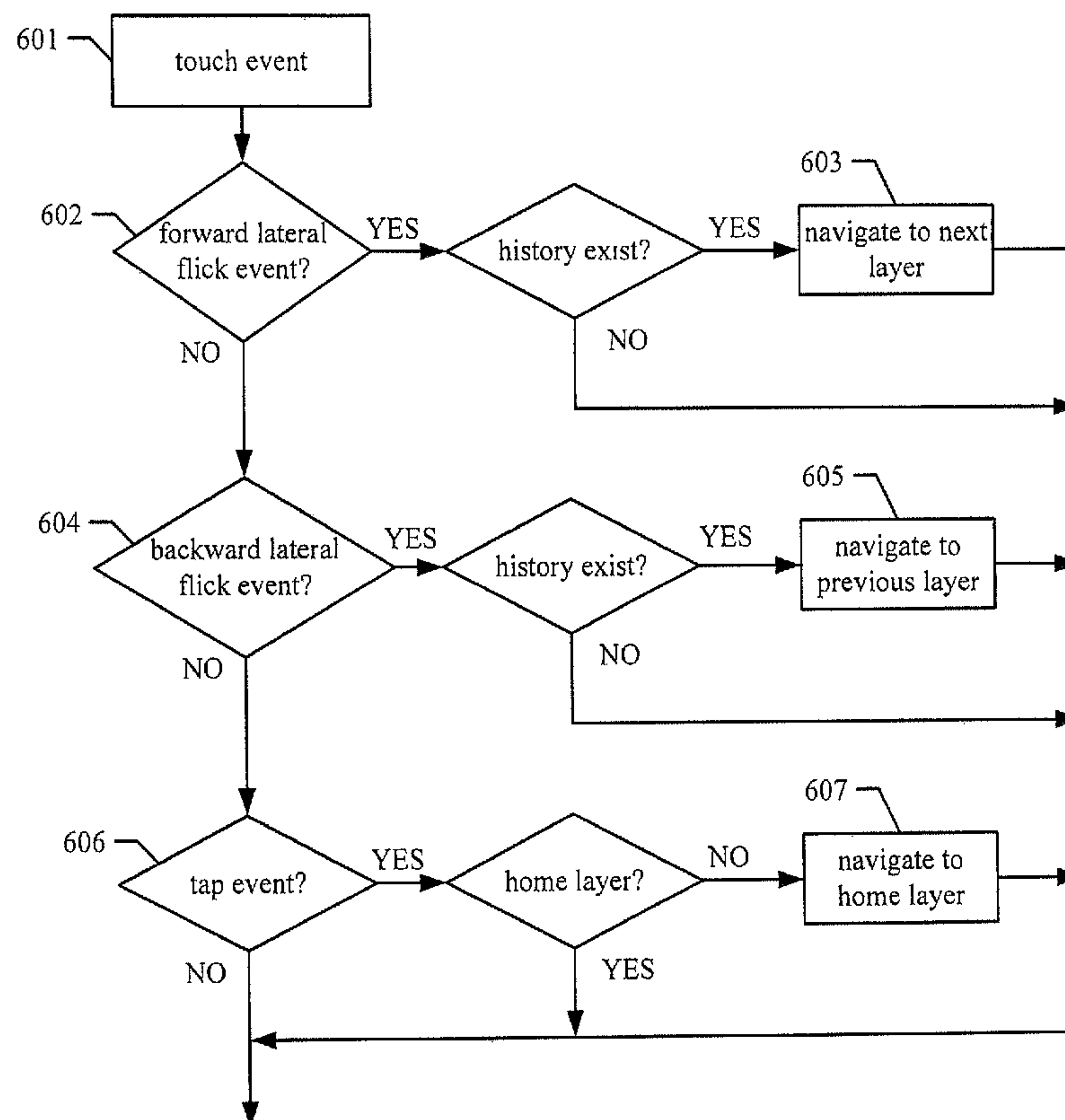




(86) Date de dépôt PCT/PCT Filing Date: 2012/09/13
(87) Date publication PCT/PCT Publication Date: 2013/04/25
(45) Date de délivrance/Issue Date: 2016/01/05
(85) Entrée phase nationale/National Entry: 2014/04/16
(86) N° demande PCT/PCT Application No.: US 2012/054973
(87) N° publication PCT/PCT Publication No.: 2013/058896
(30) Priorité/Priority: 2011/10/17 (US13/275,095)

(51) Cl.Int./Int.Cl. *G06F 3/01* (2006.01),
G06F 3/041 (2006.01), *G06F 3/048* (2013.01),
G06F 3/14 (2006.01), *G06F 9/44* (2006.01)
(72) Inventeurs/Inventors:
PAPAKIPOS, MATTHEW NICHOLAS, US;
CAHILL, MATTHEW, US
(73) Propriétaire/Owner:
FACEBOOK, INC., US
(74) Agent: STIKEMAN ELLIOTT S.E.N.C.R.L.,SRL/LLP

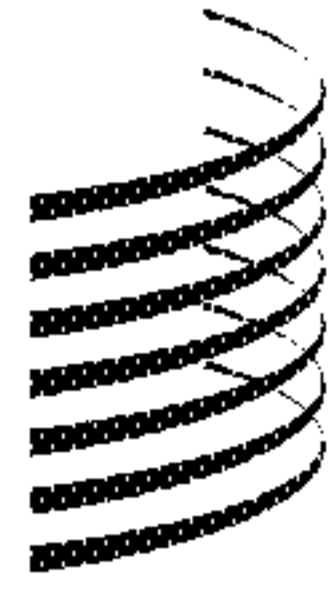
(54) Titre : APPLICATIONS DE NAVIGATION A L'AIDE D'UN PAVE TACTILE LATERAL
(54) Title: NAVIGATING APPLICATIONS USING SIDE-MOUNTED TOUCHPAD



(57) Abrégé/Abstract:

In one embodiment, a user of a mobile device navigates between application user interface layers by using touch inputs to the mobile device's side-mounted touchpad.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau(43) International Publication Date
25 April 2013 (25.04.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/058896 A1

(51) International Patent Classification:

G06F 3/01 (2006.01) G06F 3/14 (2006.01)
G06F 3/048 (2013.01) G06F 9/44 (2006.01)
G06F 3/041 (2006.01)

(21) International Application Number:

PCT/US2012/054973

(22) International Filing Date:

13 September 2012 (13.09.2012)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

13/275,095 17 October 2011 (17.10.2011) US

(71) Applicant (for all designated States except US): **FACEBOOK, INC.** [US/US]; 1601 Willow Road, Menlo Park, CA 94025 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **PAPAKIPOS, Matthew, Nicholas** [US/US]; 1601 Willow Road, Menlo Park, CA 94025 (US). **CAHILL, Matthew** [US/US]; 1601 Willow Road, Menlo Park, CA 94025 (US).(74) Agent: **CHOI, Hogene, L.**; Baker Botts L.L.P., 2001 Ross Avenue, Suite 600, Dallas, TX 75201 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

[Continued on next page]

(54) Title: NAVIGATING APPLICATIONS USING SIDE-MOUNTED TOUCHPAD

(57) Abstract: In one embodiment, a user of a mobile device navigates between application user interface layers by using touch inputs to the mobile device's side-mounted touchpad.

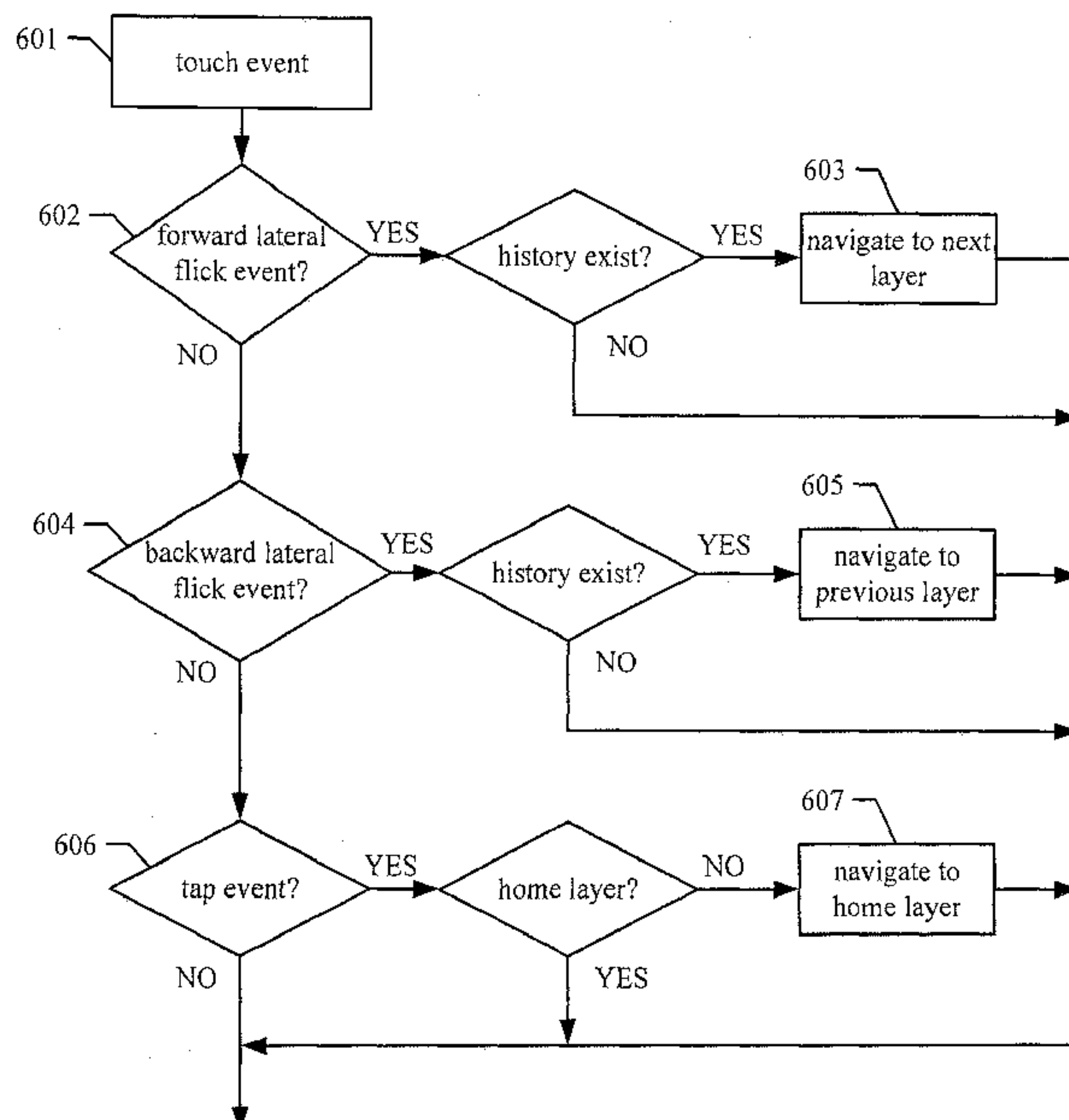


FIGURE 6

WO 2013/058896 A1



TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

NAVIGATING APPLICATIONS USING SIDE-MOUNTED TOUCHPAD

TECHNICAL FIELD

The present disclosure relates generally to touch-based user interfaces, and more particularly to, navigating between user interface layers of an application hosted by a computing device having a side-mounted touchpad.

5

BACKGROUND

A touchpad is an input device including a surface that detects touch-based inputs of users. A touch screen is an electronic visual display that detects the presence and location of user touch inputs. Mobile devices such as a mobile phone, a tablet
10 computer, and a laptop computer often incorporate a touch screen or a touchpad to facilitate user interactions with application programs running on the mobile device.

SUMMARY

Particular embodiments relate to touch-based user interfaces that allow a user of a
15 computing device to navigate between application user interface layers by using touch inputs to a side-mounted touchpad. These and other features, aspects, and advantages of the disclosure are described in more detail below in the detailed description and in conjunction with the following figures.

20 BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates an example touch screen of a mobile phone that hosts a browser client displaying a web page.

Figure 2 illustrates an example processing stack of a mobile device with touch-based input device(s).

25 Figure 3A-3C illustrate an example method of navigating between user interface layers of an application hosted on a touch-screen device.

Figure 4 illustrates an example application user interface hierarchy of an application hosted on a touch-screen device.

Figure 5 illustrates a front view and a side view of an example mobile device with a

front-mounted touch screen and a side-mounted touchpad.

Figure 5A illustrates another example of the mobile device of Figure 5.

Figure 5B illustrates an example mobile device with a back-mounted touch surface.

Figures 5C-5F illustrate example touch events associated with the one or more side-mounted touchpads of the example mobile device of Figure 5.

Figures 5G-5I illustrate example touch events associated with a back-mounted touch surface of the example mobile device of Figure 5B.

Figure 6 illustrates an example method of navigating between user interface layers of an application using side-mounted touchpads.

Figure 7 illustrates an example mobile device platform.

DETAILED DESCRIPTION

The invention is now described in detail with reference to a few embodiments thereof as illustrated in the accompanying drawings. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. It is apparent, however, to one skilled in the art, that the present disclosure may be practiced without some or all of these specific details. In other instances, well known process steps and/or structures have not been described in detail in order not to unnecessarily obscure the present disclosure. In addition, while the disclosure is described in conjunction with the particular embodiments, it should be understood that this description is not intended to limit the disclosure to the described embodiments. To the contrary, the description is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the disclosure as defined by the appended claims.

A touchpad is an input device including a surface that detects touch-based inputs of users. Similarly, a touch screen is an electronic visual display that detects the presence and location of user touch inputs. So-called dual touch or multi-touch displays or touchpads refer to devices that can identify the presence, location and movement of more than one touch input, such as two or three finger touches. A system incorporating one or more touch-based input devices may monitor one or more touch-sensitive surfaces for one or more touch or near touch inputs from a user. When one

or more such user inputs occur, the system may determine the distinct area(s) of contact and identify the nature of the touch or near touch input(s) via geometric features and geometric arrangements (e.g., location, movement), and determine if they correspond to various touch events (e.g., tap, drag, swipe, pinch). These touch events may then be processed by handler functions that register or subscribe as listeners to such events, as illustrated in Figure 1. Figure 1 illustrates an example touch screen of a mobile phone that hosts a browser client displaying a web page. In the example of Figure 1, touch screen 101 of mobile phone 100 displays an HTML/JavaScript code snippet displaying a text string “Tap on this text”, as listed below.

10

```

<html>
<!-- pseudo-code to include a touch event listener from a touch
events library
library TouchEventsLibrary
function onTapEvent
-->
<body>
<h1  onTapEvent="this.innerHTML='Boo!'">Click  on  this
text</h1>
20  </body>
    </html>

```

20

As a user taps on the text string “Tap on this text.” (102), a touch event listener “onTouchEvent” can trigger an action of changing the text string from “Tap on this text.” to “Boo!” (103).

25

Recognition of touch events by a system with one or more touch-based input devices--i.e., identifying one or more touch inputs by a user and determining corresponding touch event(s)-- may be implemented by a combination of hardware, software, and/or firmware (or device drivers). Figure 2 illustrates an example processing stack of a mobile device (e.g., a smart phone) with touch-based input device(s). Hardware layer 201 can include one or more processors and various hardware input/output devices such as camera, communication interface, and touch-based input device (e.g., touch screen, touchpad). Drivers layer 202 includes one or more drivers that communicate

30

and control hardware layer 200, for example, a driver receiving and processing touch input signals generated by a touch-screen display. Operating system 203 runs computing programs and manages hardware layer 201 via one or more drivers in driver layer 202. Libraries 204 include one or more libraries used by one or more application programs in applications 205 (e.g., web browser, address book, etc.). For example, touch events library 210 can contain codes that interpret touch inputs to touch events or gestures, and a web browser application program can access touch event library 210 (e.g., via function calls) and process a web page with touch event handlers embedded within the page, as illustrated in Figure 1 and in the HTML/JavaScript code snippet above.

User interface layers of application programs hosted by a computing device, including user interface layers of the computing device's operating system, are generally arranged in a hierarchy, such as a hierarchical tree structure. A root node of the tree structure may correspond to a home user interface layer or home screen of an application, including icons, buttons, pull-down menus, hypertext links or other controls that allow users to navigate to child user interface layers. Child user interface layers may include icons or buttons that allow users to navigate back to parent user interface layers by selecting an icon or button. Many such buttons are displayed on the top of a menu bar, but there are a lot of exceptions.

With a touch-screen device, a user can often navigate between user interface layers of an application hosted on the touch-screen device by touching one or more selectable elements within the application's user graphic interface displayed with the touch-screen, as illustrated in Figures 3A-3C. The particular application shown in Figures 3A-3C is a Facebook(r) client hosted by a mobile phone with a touch screen (e.g., an iPhone(r) smartphone offered by Apple, Inc. of Cupertino, California). When a user launches the application, the application shows a home screen 301 (Figure 3A), or an interface displaying home user interface layer of the application. The home screen 301 may include user interface elements such as icons or buttons for navigating to other layers, for example, News Feed icon 310, Photos icon 330, or Friends icon 340. Figure 4 illustrates an example application user interface hierarchy 400 of the application illustrated in Figures 3A-3C. The application user interface hierarchy includes home layer 401 and one or more sub-layers 402, such as News Feed 410,

Profile 420, Photos 430, Friends 440, and the like. In addition, Photos layer 430 itself has a sub-layer of Album list 432 which in turn has a sub-layer Photo list 434, and so on. Not all layers have sub-layers, such as News Feed content 412, Self Profile content 422, "Photo 1" 436, which are leaf layers. In the example of Figure 4, the nodes of the hierarchy represent different user interface layers provided by the application, while the edges between the nodes illustrate a potential navigation flow. Additionally, the application may allow for navigation among the nodes in a way that does not observe a hierarchical model, such as navigating directly between sibling node layers, traversing more than one hop directly between layers, and the like.

For example, from home layer 401, the user can navigate to Friends layer 440 by touching Friends icon 340 within home screen 301, the current screen will change to a screen 341 for Friends layer 440, as illustrated in Figure 3B. The screen 341 for Friends layer 440 comprises a list of friends wherein each friend is represented by a selectable icon (e.g., a profile picture). The user can navigate to a particular friend layer "Laura" 442 by touching a corresponding icon (342), the current screen will change to a screen for the particular friend (343), as illustrated in Figure 3C. From the particular friend layer "Laura" 442 or from the screen for the particular friend 343 illustrated in Figure 3C, the user can look up additional information (e.g., an email address, a phone number) of the particular friend by touching "info" icon 344, navigate back to Friends layer by touching "Friends" icon 345, or navigating to home layer by touching home icon 346.

Particular embodiments herein relate to a computing device (such as a mobile phone, netbook, smartphone, tablet, or other portable device) with a touch screen and one or more side-mounted touchpads and methods of allowing users to use the one or more side-mounted touchpads to navigate between application user interface layers. Particular embodiments can improve user experience associated with mobile devices as the side-mounted touchpad(s) can offload user interaction to the side-mounted touchpad(s) and yield better usage of the touch screen (e.g., a less congested graphic user interface with less navigation elements or icons). Figure 5 illustrates a front view and a side view of an example mobile device with a front-mounted touch screen and a side-mounted touchpad. In particular embodiments, mobile device 500 may comprise a housing with multi-touch touch screen 501 disposed on a front face of the housing.

The mobile device 500 may also include a side-mounted multi-touch touchpad 502 and a side-mounted single-touch touchpad 503, both disposed on a lateral face or edge of the device 500. In particular embodiments, mobile device 500 may include hardware and/or software that supports or implements a variety of functions. For example, mobile device 500 may support telephony functions, chat and/or email functions. Mobile device 500 may also support network data communications and include a web browser for accessing and displaying web pages. Mobile device 500 may also support or incorporate, a Wi-Fi base station functions, a digital media player functions, and/or a gaming device functions. In one embodiment, the side-mounted touchpad 503 may be replaced by a clickable button or keypad device. In another embodiment, the side-mounted touchpad 503 may be a multi-touch touchpad. In some implementations, the touchpad 402 may be a single- or multi-touch device. In some embodiments, side-mounted touchpad 502 may comprise a slightly concave multi-touch surface, as illustrated in Figure 5A. The touch screen 501 and side-mounted touchpad 503 may be single-touch, dual-touch or multi-touch devices. In addition, implementations of the invention can operate without a touch screen device, relying instead on a regular display device and a pointer device, such as a trackball or trackpad. In other embodiments, mobile device 500 may include a back-mount touch surface 505 on a back-side of mobile device 500. The back-mounted touch 505 may cover substantially all or a portion of a back side of mobile device 500, as illustrated in Figure 5B. The back-mounted touch surface 505 may comprise a multi-touch touchpad or a multi-touch touch screen.

Mobile device 500 may recognize touch inputs, and determine one or more corresponding touch events or gestures. One or more applications hosted on mobile device 500 may be configured to register a handler function that responds to the one or more touch events. In particular embodiments, mobile device 500 may recognize one or more user touch inputs performed on touch screen 501, touchpad 502, touchpad 503, and/or back-mounted touch surface 505, and determine one or more corresponding touch events. In particular embodiments, mobile device 500 may determine a forward lateral flick event associated with touchpad 502, as illustrated in Figure 5C. In the example of Figure 5C, a user touches touchpad 502 in a quick motion in forward lateral direction (indicated by the arrow 510), and a gesture

recognition library of mobile device 500 can access data generated by touchpad 502 and determine the touch input corresponding to a forward lateral flick event or gesture associated with touchpad 502. In particular embodiments, mobile device 500 may determine a backward lateral flick event associated with touchpad 502, as illustrated in Figure 5D. In the example of Figure 5D, a user touches touchpad 502 in a quick motion in backward lateral direction (indicated by the arrow 512), and a gesture recognition library of mobile device 500 can access data generated by touchpad 502 and determine the touch input corresponding to a backward lateral flick event or gesture associated with touchpad 502.

10 In particular embodiments, mobile device 500 may identify a tap event. In some embodiments, mobile device 500 may determine a tap event associated with touchpad 502, as illustrated in Figure 5E. In the example of Figure 5E, a user taps or strikes lightly on touchpad 502 (indicated by the arrow 514), and a gesture recognition library of mobile device 500 can interpret the user's touch input and identify the touch

15 input corresponding to a tap event. In other embodiments, mobile device 500 may identify a tap event associated with touchpad 503, as illustrated in Figure 5F. In the example of Figure 5F, a user taps or strikes lightly on touchpad 503 (indicated by the arrow 516), and a gesture recognition library of mobile device 500 can interpret the user's touch input and identify the touch input corresponding to a tap event. In one

20 embodiments, mobile device 500 may identify a tap event associated with touchpad 503 if touchpad 503 is a clickable button and a user clicks on the clickable button. In other embodiments, mobile device 500 may identify touch events associated with back-mounted touch surface 505, as illustrated in Figures 5G-5I. In the example of Figure 5G, a user touches back-mounted touch surface 505 in a quick motion in an

25 upward direction (indicated by the arrow 531). A gesture recognition library of mobile device 500 can interpret the user's touch input and identify the touch input corresponding to an upward flick event or gesture associated with back-mounted touch surface 505. In the example of Figure 5H, a user touches back-mounted touch surface 505 in a quick motion in a downward direction (indicated by the arrow 532).

30 A gesture recognition library of mobile device 500 can interpret the user's input and identify the touch input corresponding to a downward flick event or gesture associated with back-mounted touch surface 505. In the example of Figure 5I, a user taps or

strikes lightly on back-mounted touch surface 505 (as indicated by the arrow 533). A gesture recognition library of mobile device 500 can interpret the user's input and identify the touch input corresponding to a tap event associated with back-mounted touch surface 505.

- 5 In contrast to navigating between user interface layers of an application by touch inputs to a touch screen as illustrated in the example in Figures 3A-3C, Figure 6 illustrates an example method of navigating between user interface layers of an application using side-mounted touchpads. Specifically, the example method of Figure 6 may enable a user to navigate between user interface layers by flicking
- 10 across a side-mounted touchpad, to jump to the application's home screen by tapping or clicking a side-mounted "home button". As a listener for touch events (601), the handler function can, responsive to a touch event, cause an application to navigate to a different user interface layer. In particular embodiments, when a touch event occurs (601), the handler function may determine a current application layer corresponding
- 15 to the touch event. In particular embodiments, the application may store a navigation history that tracks the layers to which the user has navigated. In particular embodiments, the handler function may determine if the touch event is a forward lateral flick event (602). In particular embodiments, if the touch event is a forward lateral flick event, and if the forward navigation history identifies a next layer for the
- 20 current layer, the handler function may cause the application navigate to the next layer (603). In particular embodiments, the handler function may determine if the touch event is a backward lateral flick event (604). In particular embodiments, if the touch event is a backward lateral flick event, and if the backward navigation history identifies a previous layer for the current layer, the handler function may cause the
- 25 application navigate to the previous layer (605). In particular embodiments, the handler function may determine if the touch event is a tap event in connection with input device 503 (606). In particular embodiments, if the touch event is a tap event, and if the current layer is not a home layer, the handler function may cause the application navigate to the home layer (607).
- 30 Additionally, particular embodiments may enable navigating between user interface layers of an application using the back-mounted touch surface described earlier. For example, as a listener for touch events, the handler function can, responsive to a touch

event, cause an application to navigate to a different user interface layer. When a touch event occurs, the handler function may determine a current application layer corresponding to the touch event. The application may store a navigation history that tracks the layers to which the user has navigated. The handler function may
5 determine if the touch event is an upward flick event associated with back-mounted touch surface 505. If the touch event is an upward flick event associated with back-mounted touch surface 505, and if the forward navigation history identifies a next layer for the current layer, the handler function may cause the application to navigate to the next layer. The handler function may determine if the touch event is a
10 downward flick event associated with back-mounted touch surface 505. If the touch event is a downward flick event associated with back-mounted touch surface 505, and if the backward navigation history identifies a previous layer for the current layer, the handler function may cause the application to navigate to the previous layer. The handler function may determine if the touch event is a tap event associated with back-mounted touch surface 505. If the touch event is a tap event associated with back-mounted touch surface 505, and if the current layer is not a home layer, the handler function may cause the application to navigate to the home layer.

The forward and back gestures, as well as the home control input, described above can be utilized in connection with a variety of applications and computing devices.
20 For example, as discussed above, the forward and back gestures, as well as the home control input, may be used in connection with a special-purpose client application in order to facilitate navigation of the various application layers. In such an implementation, the home input causes navigation back to the home or root layer of the application. The forward and back gestures, as well as the home control input,
25 may also be used in connection with a browser client to facilitate navigation of a series of web pages provided by one or more domains. In such an implementation, the home input may cause navigation back to a home page, while the forward and back gestures may implement corresponding forward and backward navigation functions common to browser client applications. In addition, the forward and back
30 gestures, as well as the home control input, can be used in connection with a graphical user interface shell of an operating system. In such an implementation, the home input may cause navigation back to the home screen of the operating system, while

the forward and back gestures may implement corresponding forward and backward navigation functions across various layers provided by the operating system shell.

The application and functionality described above can be implemented as a series of instructions stored on a computer-readable storage medium that, when executed,
5 cause a programmable processor to implement the operations described above. While the mobile device 500 may be implemented in a variety of different hardware and computing systems, FIGURE 7 shows a schematic representation of the main components of an example computing platform of a client or mobile device, according to various particular embodiments. In particular embodiments, computing
10 platform 702 may comprise controller 704, memory 706, and input output subsystem 710. In particular embodiments, controller 704 which may comprise one or more processors and/or one or more microcontrollers configured to execute instructions and to carry out operations associated with a computing platform. In various embodiments, controller 704 may be implemented as a single-chip, multiple chips
15 and/or other electrical components including one or more integrated circuits and printed circuit boards. Controller 704 may optionally contain a cache memory unit for temporary local storage of instructions, data, or computer addresses. By way of example, using instructions retrieved from memory, controller 704 may control the reception and manipulation of input and output data between components of
20 computing platform 702. By way of example, controller 704 may include one or more processors or one or more controllers dedicated for certain processing tasks of computing platform 702, for example, for 2D/3D graphics processing, image processing, or video processing.

Controller 704 together with a suitable operating system may operate to execute
25 instructions in the form of computer code and produce and use data. By way of example and not by way of limitation, the operating system may be Windows-based, Mac-based, or Unix or Linux-based, Symbian-based, or Android-based among other suitable operating systems. The operating system, other computer code and/or data may be physically stored within memory 706 that is operatively coupled to controller
30 704.

Memory 706 may encompass one or more storage media and generally provide a place to store computer code (e.g., software and/or firmware) and data that are used

by computing platform 702. By way of example, memory 706 may include various tangible computer-readable storage media including Read-Only Memory (ROM) and/or Random-Access Memory (RAM). As is well known in the art, ROM acts to transfer data and instructions uni-directionally to controller 704, and RAM is used typically to transfer data and instructions in a bi-directional manner. Memory 706 may also include one or more fixed storage devices in the form of, by way of example, hard disk drives (HDDs), solid-state drives (SSDs), flash-memory cards (e.g., Secured Digital or SD cards, embedded MultiMediaCard or eMMD cards), among other suitable forms of memory coupled bi-directionally to controller 704. Information may also reside on one or more removable storage media loaded into or installed in computing platform 702 when needed. By way of example, any of a number of suitable memory cards (e.g., SD cards) may be loaded into computing platform 702 on a temporary or permanent basis.

Input output subsystem 710 may comprise one or more input and output devices operably connected to controller 704. For example, input-output subsystem may include keyboard, mouse, one or more buttons, thumb wheel, and/or display (e.g., liquid crystal display (LCD), light emitting diode (LED), Interferometric modulator display (IMOD), or any other suitable display technology). Generally, input devices are configured to transfer data, commands and responses from the outside world into computing platform 702. The display is generally configured to display a graphical user interface (GUI) that provides an easy to use visual interface between a user of the computing platform 702 and the operating system or application(s) running on the mobile device. Generally, the GUI presents programs, files and operational options with graphical images. During operation, the user may select and activate various graphical images displayed on the display in order to initiate functions and tasks associated therewith. Input output subsystem 710 may also include touch based devices such as touchpad and touch screen. A touchpad is an input device including a surface that detects touch-based inputs of users. Similarly, a touch screen is a display that detects the presence and location of user touch inputs. Input output system 710 may also include dual touch or multi-touch displays or touchpads that can identify the presence, location and movement of more than one touch inputs, such as two or three finger touches.

In particular embodiments, computing platform 702 may additionally comprise audio subsystem 712, camera subsystem 712, wireless communication subsystem 716, sensor subsystems 718, and/or wired communication subsystem 720, operably connected to controller 704 to facilitate various functions of computing platform 702.

5 For example, Audio subsystem 712, including a speaker, a microphone, and a codec module configured to process audio signals, can be utilized to facilitate voice-enabled functions, such as voice recognition, voice replication, digital recording, and telephony functions. For example, camera subsystem 712, including an optical sensor (e.g., a charged coupled device (CCD), or a complementary metal-oxide
10 semiconductor (CMOS) image sensor), can be utilized to facilitate camera functions, such as recording photographs and video clips. For example, wired communication subsystem 720 can include a Universal Serial Bus (USB) port for file transferring, or a Ethernet port for connection to a local area network (LAN). Additionally, computing platform 702 may be powered by power source 732.

15 Wireless communication subsystem 716 can be designed to operate over one or more wireless networks, for example, a wireless PAN (WPAN) (such as, for example, a BLUETOOTH WPAN, an infrared PAN), a WI-FI network (such as, for example, an 802.11a/b/g/n WI-FI network, an 802.11s mesh network), a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile
20 Communications (GSM) network, an Enhanced Data Rates for GSM Evolution (EDGE) network, a Universal Mobile Telecommunications System (UMTS) network, and/or a Long Term Evolution (LTE) network). Additionally, wireless communication subsystem 716 may include hosting protocols such that computing platform 702 may be configured as a base station for other wireless devices.

25 Sensor subsystem 718 may include one or more sensor devices to provide additional input and facilitate multiple functionalities of computing platform 702. For example, sensor subsystems 718 may include GPS sensor for location positioning, altimeter for altitude positioning, motion sensor for determining orientation of a mobile device, light sensor for photographing function with camera subsystem 714, temperature
30 sensor for measuring ambient temperature, and/or biometric sensor for security application (e.g., fingerprint reader). Other input/output devices may include an accelerometer that can be used to detect the orientation of the device.

In particular embodiments, various components of computing platform 702 may be operably connected together by one or more buses (including hardware and/or software). As an example and not by way of limitation, the one or more buses may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCI-X) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, a Universal Asynchronous Receiver/Transmitter (UART) interface, a Inter-Integrated Circuit (I²C) bus, a Serial Peripheral Interface (SPI) bus, a Secure Digital (SD) memory interface, a MultiMediaCard (MMC) memory interface, a Memory Stick (MS) memory interface, a Secure Digital Input Output (SDIO) interface, a Multi-channel Buffered Serial Port (McBSP) bus, a Universal Serial Bus (USB) bus, a General Purpose Memory Controller (GPMC) bus, a SDRAM Controller (SDRC) bus, a General Purpose Input/Output (GPIO) bus, a Separate Video (S-Video) bus, a Display Serial Interface (DSI) bus, an Advanced Microcontroller Bus Architecture (AMBA) bus, or another suitable bus or a combination of two or more of these.

Herein, reference to a computer-readable storage medium encompasses one or more non-transitory, tangible computer-readable storage media possessing structure. As an example and not by way of limitation, a computer-readable storage medium may include a semiconductor-based or other integrated circuit (IC) (such, as for example, a field-programmable gate array (FPGA) or an application-specific IC (ASIC)), a hard disk, an HDD, a hybrid hard drive (HHD), an optical disc, an optical disc drive (ODD), a magneto-optical disc, a magneto-optical drive, a floppy disk, a floppy disk drive (FDD), magnetic tape, a holographic storage medium, a solid-state drive (SSD), a RAM-drive, a SECURE DIGITAL card, a SECURE DIGITAL drive, a MultiMediaCard (MMC) card, an embedded MMC (eMMC) card, or another suitable computer-readable storage medium or a combination of two or more of these, where appropriate. Herein, reference to a computer-readable storage medium excludes any medium that is not eligible for patent protection under 35 U.S.C. § 101. Herein,

reference to a computer-readable storage medium excludes transitory forms of signal transmission (such as a propagating electrical or electromagnetic signal per se) to the extent that they are not eligible for patent protection under 35 U.S.C. § 101.

This disclosure contemplates one or more computer-readable storage media
5 implementing any suitable storage. In particular embodiments, a computer-readable storage medium implements one or more portions of controller 704 (such as, for example, one or more internal registers or caches), one or more portions of memory 705, or a combination of these, where appropriate. In particular embodiments, a computer-readable storage medium implements RAM or ROM. In particular
10 embodiments, a computer-readable storage medium implements volatile or persistent memory. In particular embodiments, one or more computer-readable storage media embody software. Herein, reference to software may encompass one or more applications, bytecode, one or more computer programs, one or more executables, one or more instructions, logic, machine code, one or more scripts, or source code, and
15 vice versa, where appropriate. In particular embodiments, software includes one or more application programming interfaces (APIs). This disclosure contemplates any suitable software written or otherwise expressed in any suitable programming language or combination of programming languages. In particular embodiments, software is expressed as source code or object code. In particular embodiments,
20 software is expressed in a higher-level programming language, such as, for example, C, Perl, JavaScript, or a suitable extension thereof. In particular embodiments, software is expressed in a lower-level programming language, such as assembly language (or machine code). In particular embodiments, software is expressed in JAVA. In particular embodiments, software is expressed in Hyper Text Markup
25 Language (HTML), Extensible Markup Language (XML), or other suitable markup language

The present disclosure encompasses all changes, substitutions, variations, alterations, and modifications to the example embodiments herein that a person having ordinary skill in the art would comprehend. Similarly, where appropriate, the appended claims
30 encompass all changes, substitutions, variations, alterations, and modifications to the example embodiments herein that a person having ordinary skill in the art would comprehend.

Claims

1. A method comprising:

by a computing device, detecting a touch event on a side-mounted touchpad of the computing device during display of a current user interface of an application executing on the computing device, wherein the application comprises a navigation hierarchy that comprises a home user-interface layer, a plurality of sub user-interface layers, and a plurality of user-interface branches;

by the computing device, determining a particular sub user-interface layer and user-interface branch that the current user interface corresponds to;

by the computing device, if the touch event is a forward lateral flick event and there is a next user-interface layer in the particular user-interface branch of the current user interface, then navigating to the next user-interface layer in the particular user-interface branch of the current user interface; and

by the computing device, if the touch event is a backward lateral flick event and there is a previous sub user-interface layer in the particular user-interface branch of the current user interface, then navigating to the previous user-interface layer in the particular user-interface branch of the current user interface.

2. The method of Claim 1, further comprising if the touch event is a tap event, then navigating directly to the home user-interface layer.

3. The method of Claim 1, wherein the application is a web browser.

4. The method of Claim 1, wherein the application is an operating system.

5. A device comprising:

a display;

a side-mounted touchpad;

one or more processors; and

a memory coupled to the processors comprising instructions executable by the processors, the

processors being operable when executing the instructions to:

detect a touch event on the side-mounted touchpad of the device during display of a current user interface of an application executing on the computing device, wherein the application comprises a navigation hierarchy that comprises a home user-interface layer, a plurality of sub user-interface layers, and a plurality of user-interface branches;

determine a particular sub user-interface layer and user-interface branch that the current user interface corresponds to;

if the touch event is a forward lateral flick event and there is a next user-interface layer in the particular user-interface branch of the current user interface, then navigate to the next user-interface layer in the particular user-interface branch of the current user interface; and

by the computing device, if the touch event is a backward lateral flick event and there is a previous sub user-interface layer in the particular user-interface branch of the current user interface, then navigating to the previous user-interface layer in the particular user-interface branch of the current user interface.

6. The device of Claim 5, wherein the processors are further operable when executing the instructions to if the touch event is a tap event, then navigate directly to the home user-interface layer.

7. The device of Claim 5, wherein the display is a touch screen.

8. The device of Claim 5, wherein the side-mounted touchpad comprises a concave surface.

9. 11. The device of Claim 5, wherein the application is a web browser.

10. The device of Claim 5, wherein the application is an operating system.

11. One or more computer-readable non-transitory storage media embodying software that is operable when executed to:

detect a touch event on a side-mounted touchpad of the computing device during display of a current user interface of an application executing on the computing device, wherein the application comprises a navigation hierarchy that comprises a home user-interface layer, a

plurality of sub user-interface layers, and a plurality of user-interface branches;

determine a particular sub user-interface layer and user-interface branch that the current user interface corresponds to;

if the touch event is a forward lateral flick event and there is a next user-interface layer in the particular user-interface branch of the current user interface, then navigate to the next user-interface layer in the particular user-interface branch of the current user interface; and

if the touch event is a backward lateral flick event and there is a previous sub user-interface layer in the particular user-interface branch of the current user interface, then navigate to the previous user-interface layer in the particular user-interface branch of the current user interface.

12. The media of Claim 11, wherein the software is further operable when executed to if the touch event is a tap event, then navigate directly to the home user-interface layer.

13. The media of Claim 11, wherein the application is a web browser.

14. The media of Claim 11, wherein the application is an operating system.

15. The method of Claim 1, wherein the side-mounted touchpad comprises a concave surface.

16. The media of Claim 11, wherein the side-mounted touchpad comprises a concave surface.

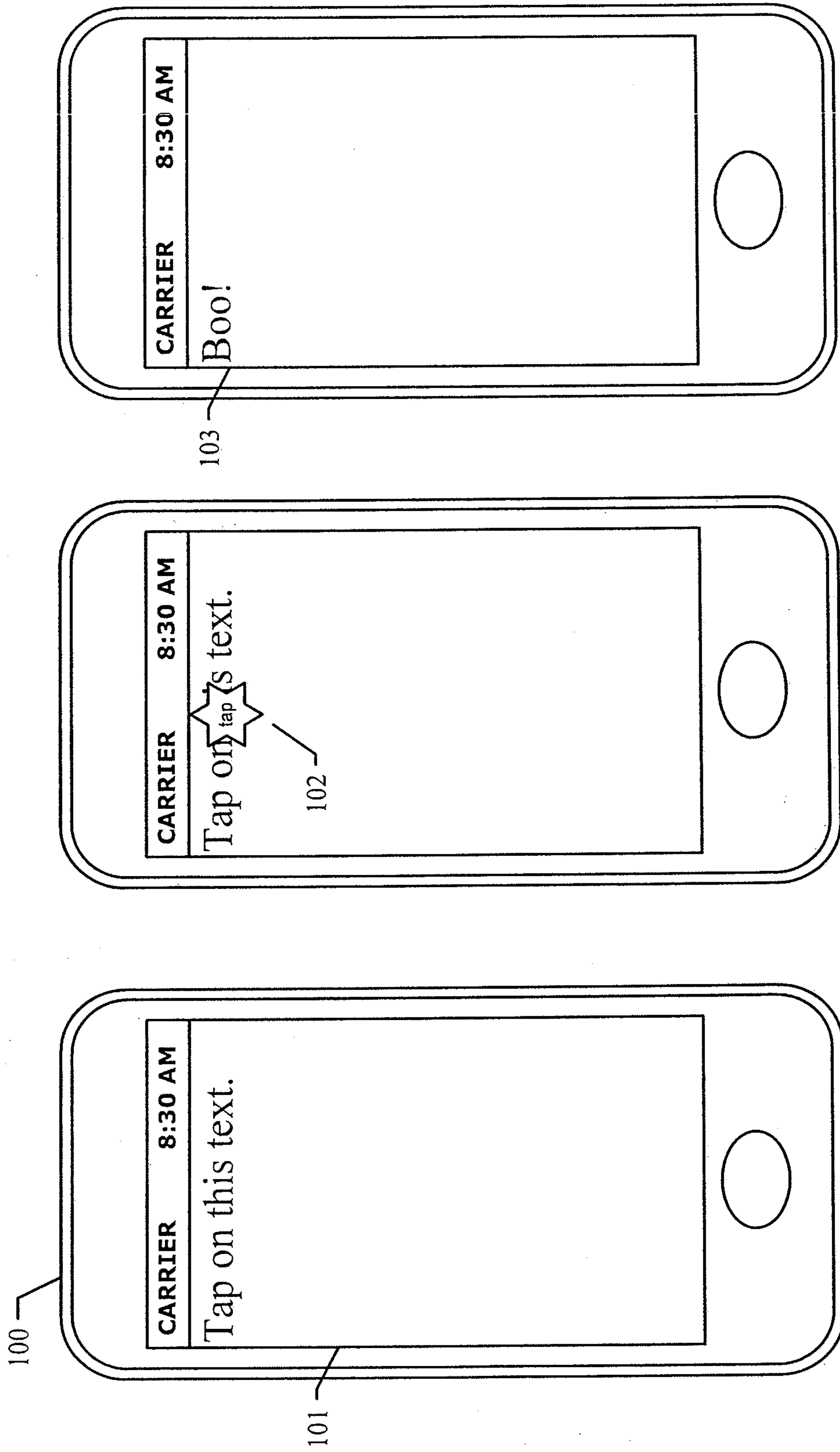


FIGURE 1

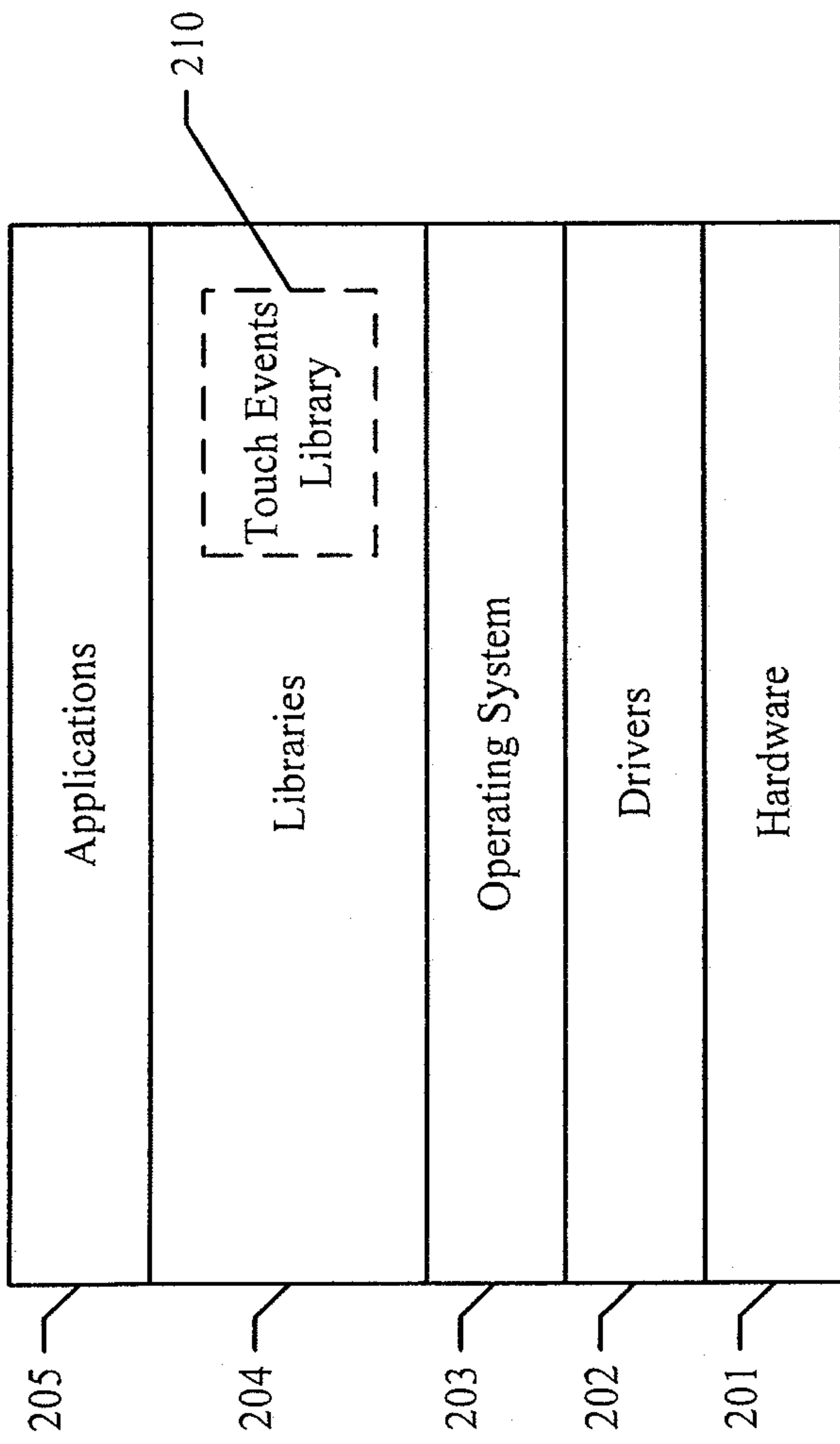


FIGURE 2

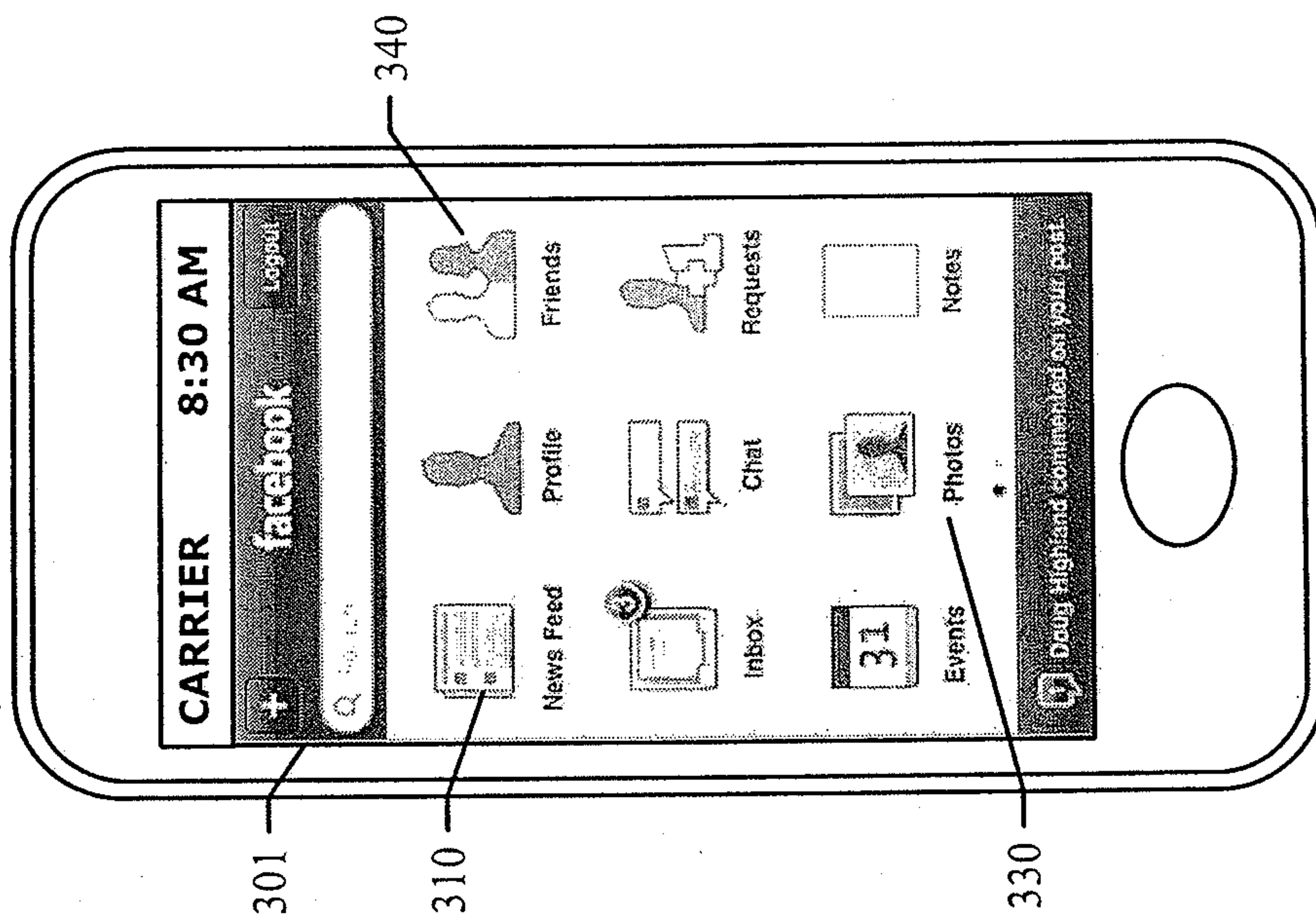


FIGURE 3A

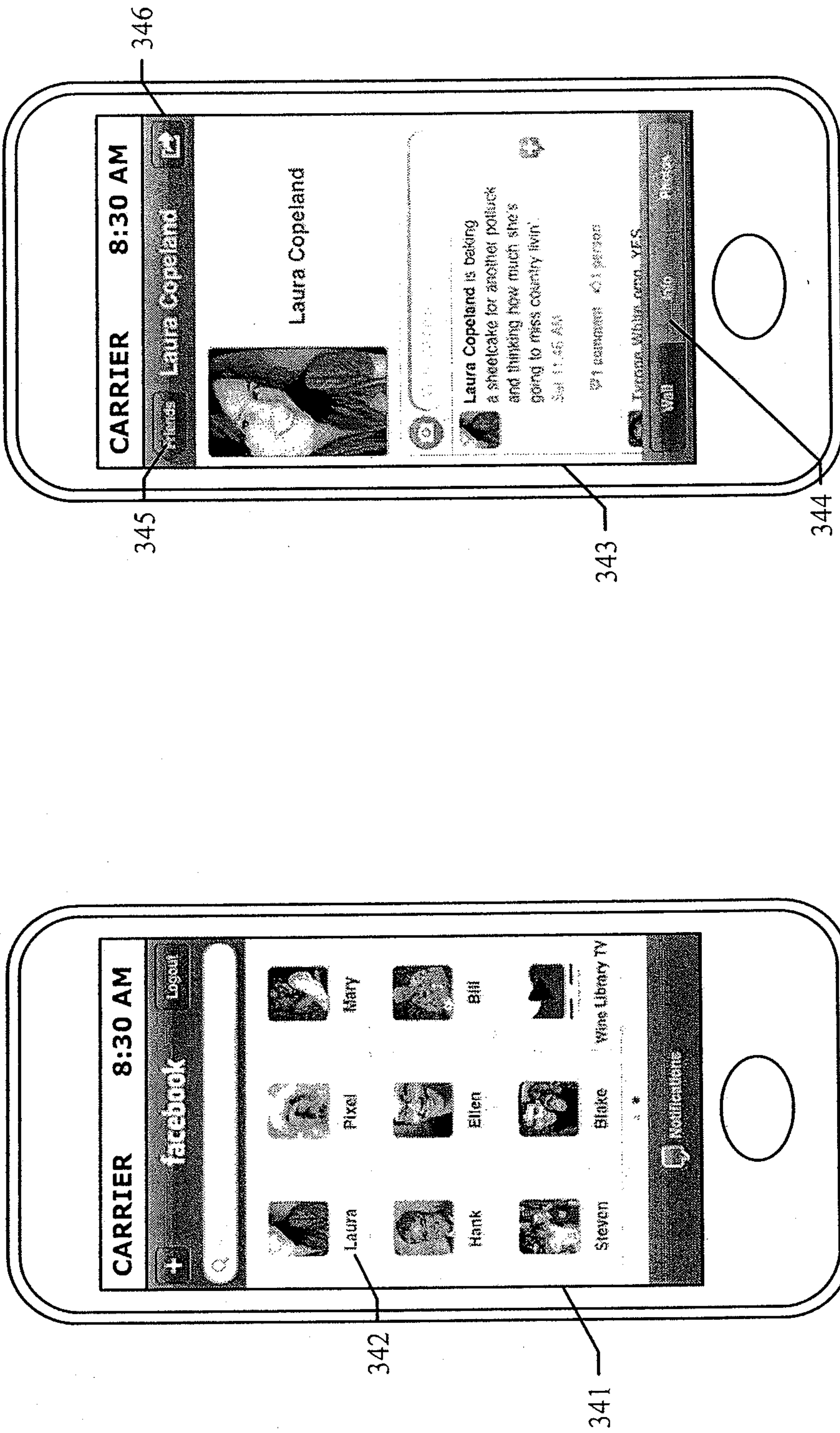


FIGURE 3B

FIGURE 3C

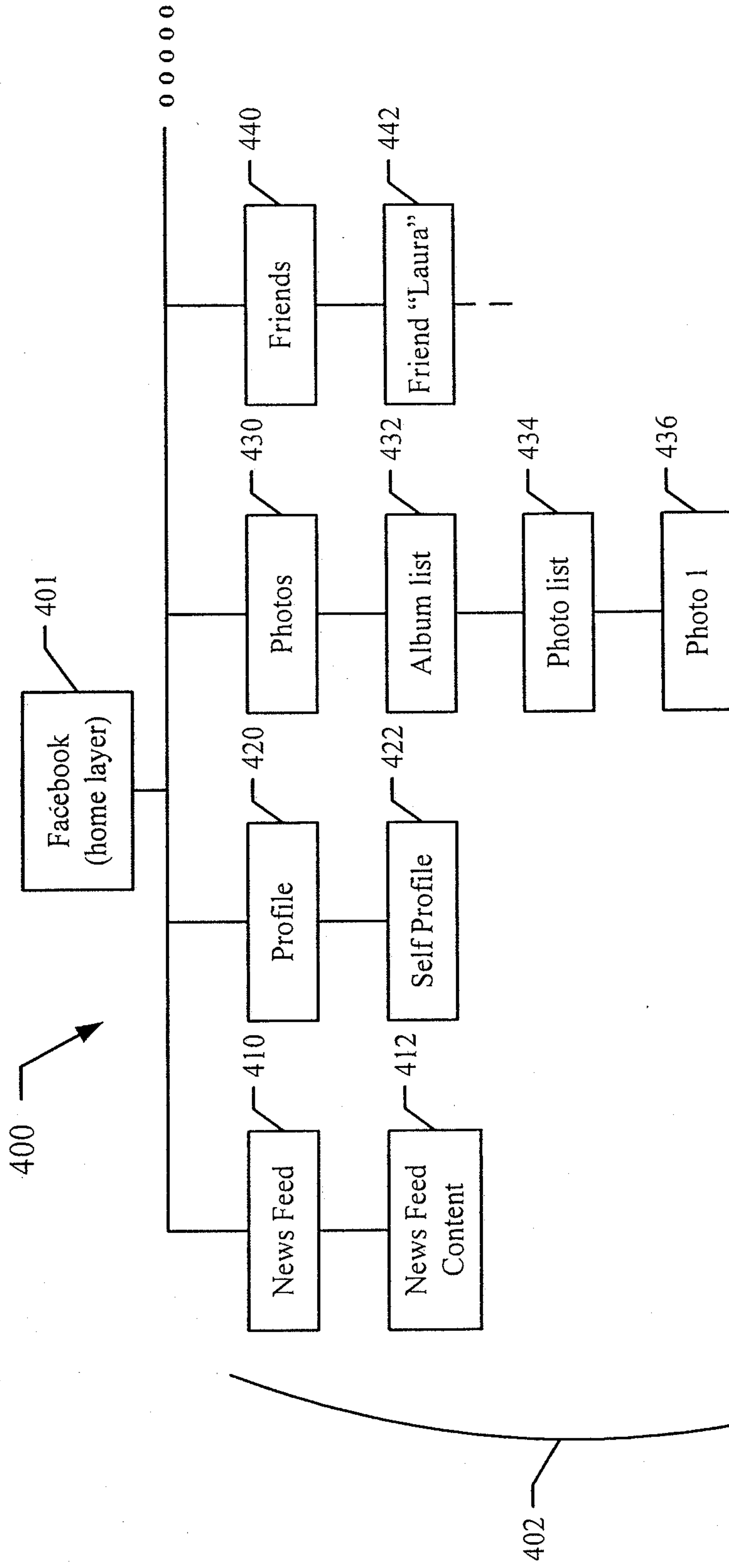


FIGURE 4

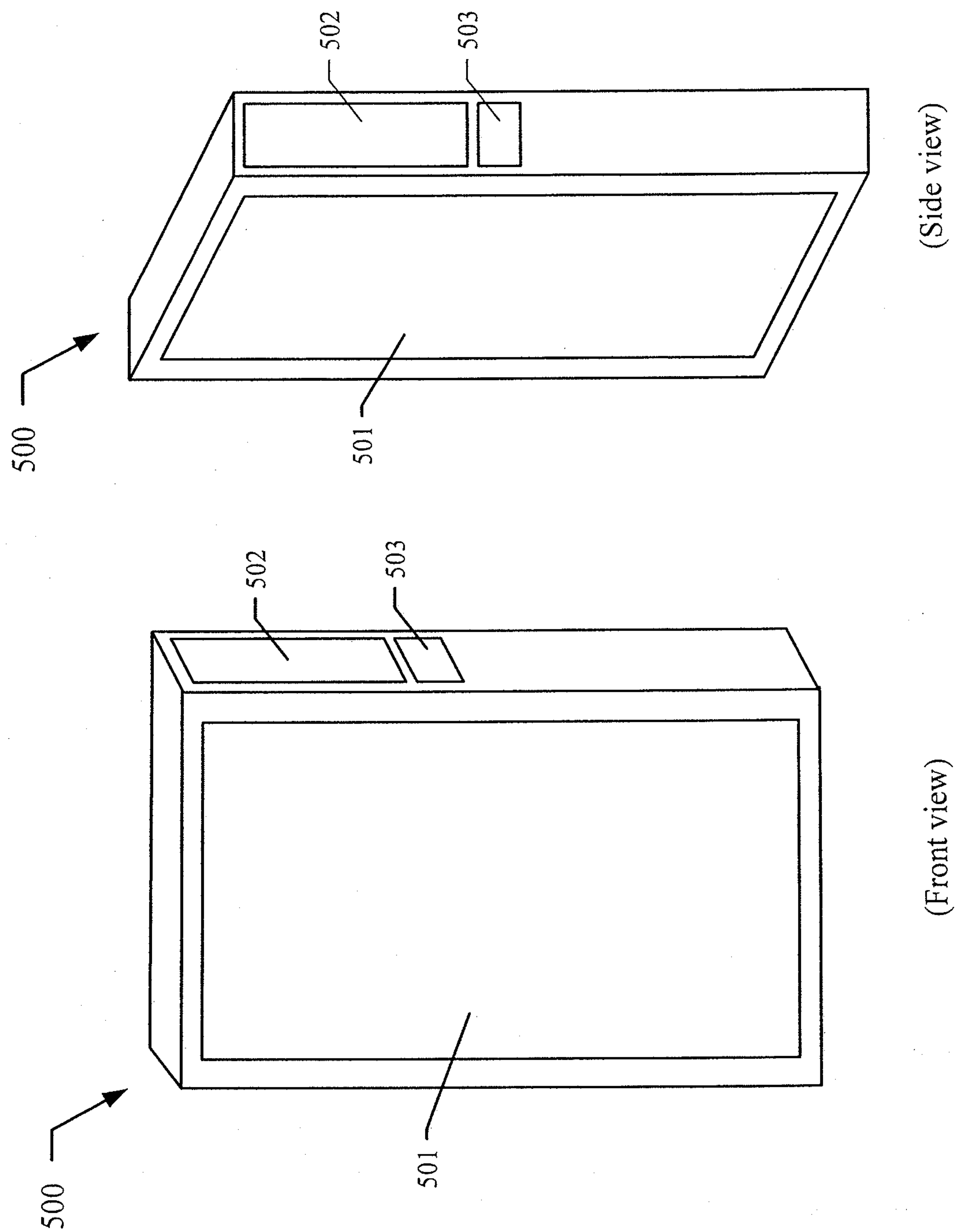


FIGURE 5

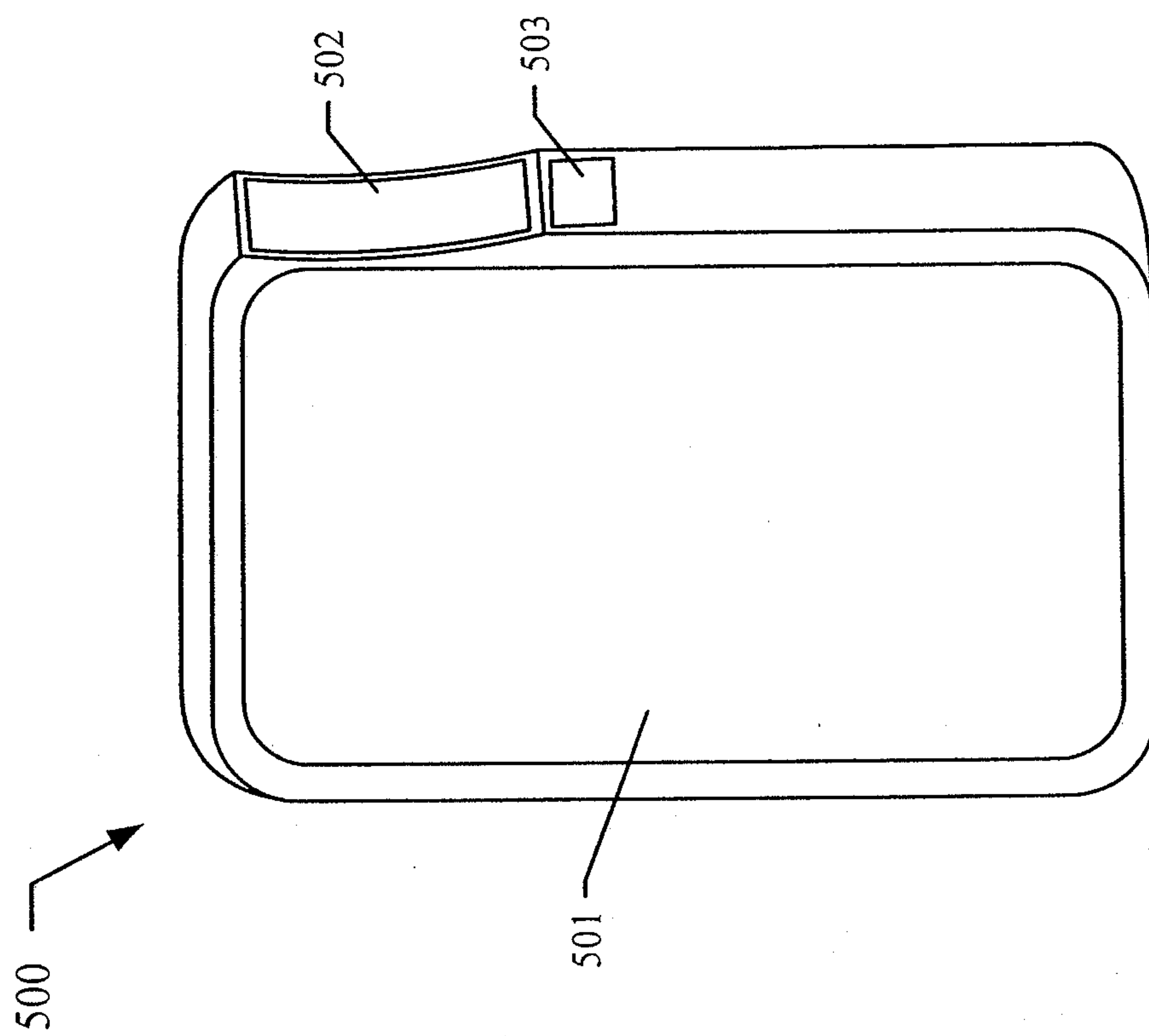


FIGURE 5A

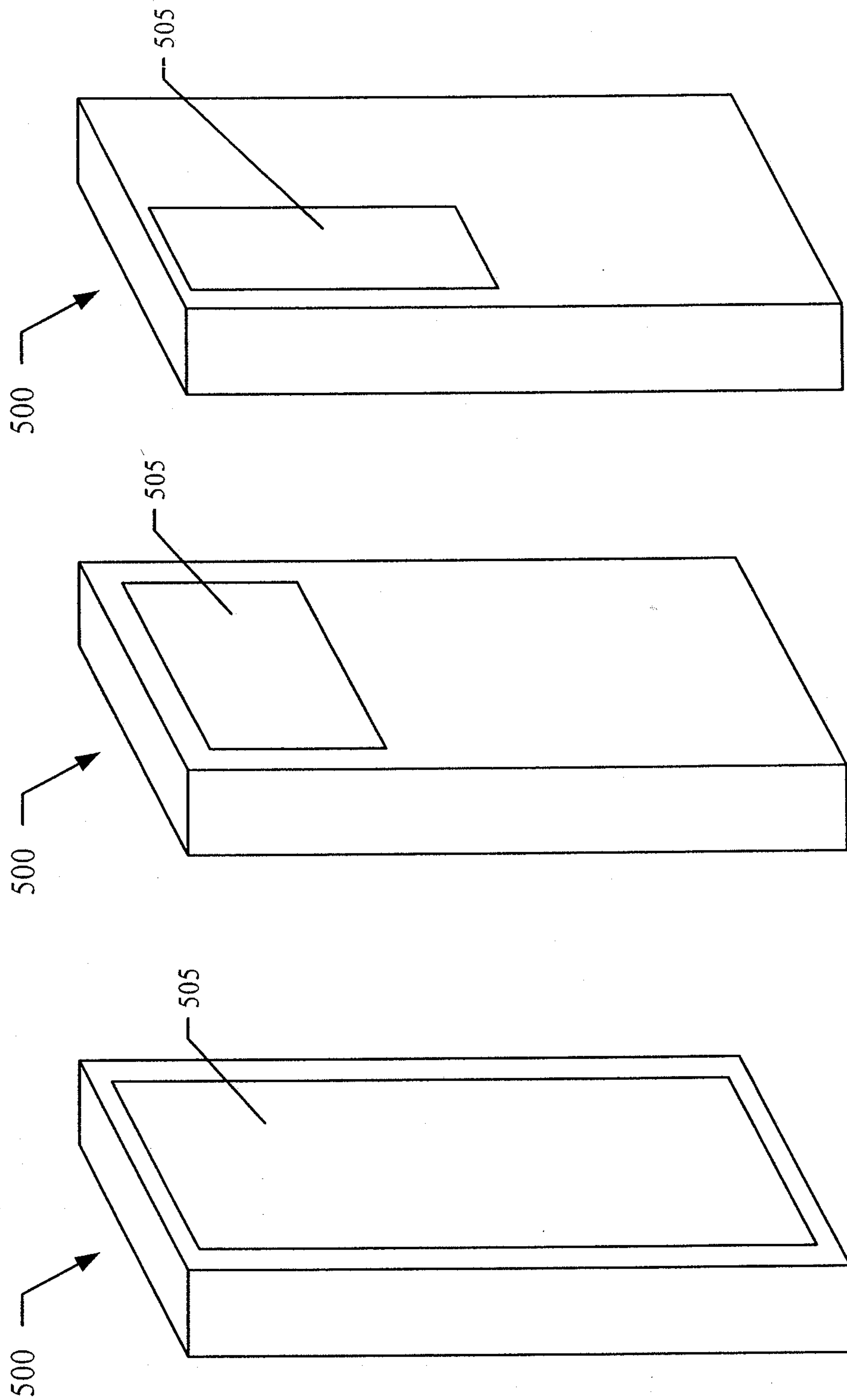
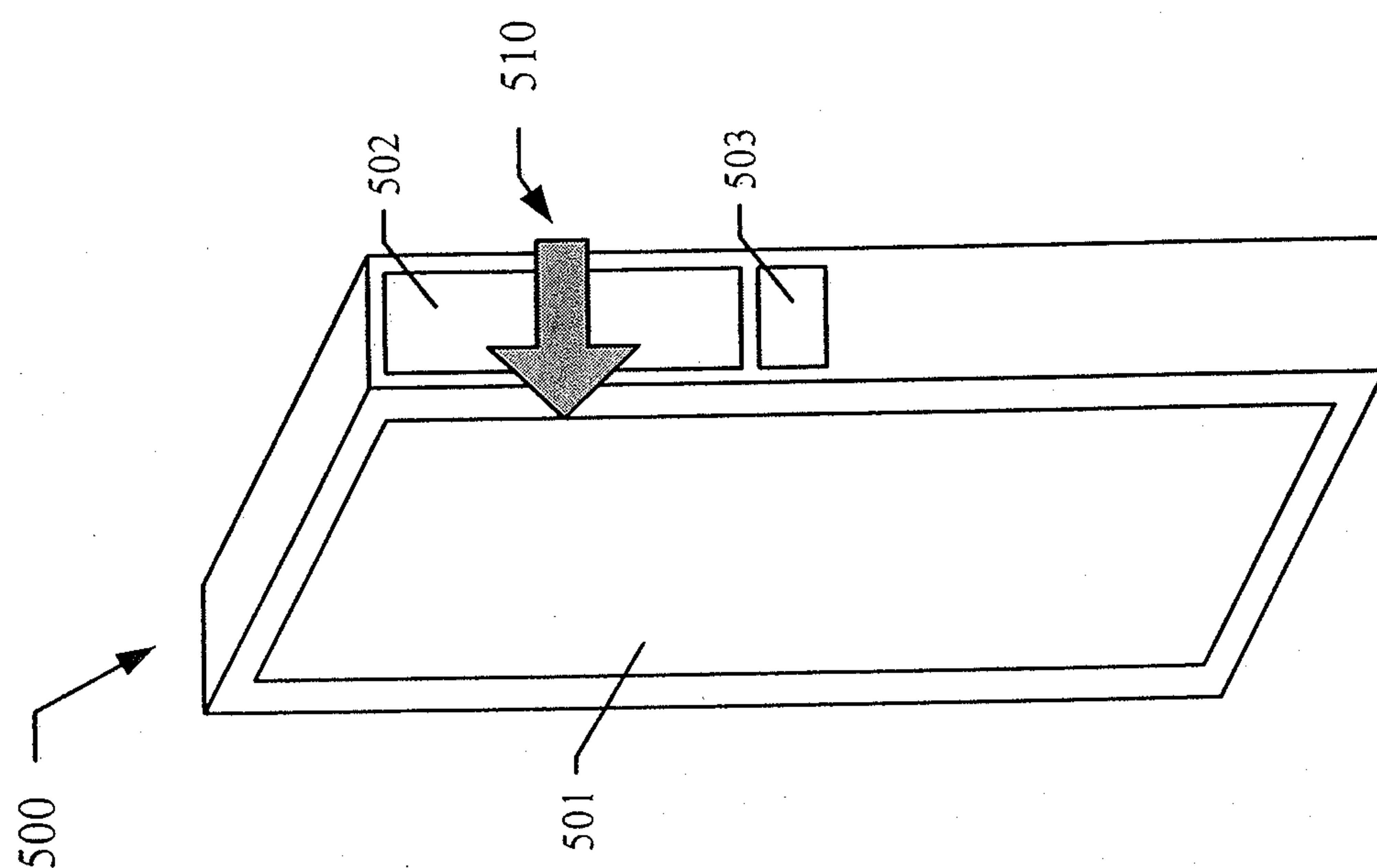
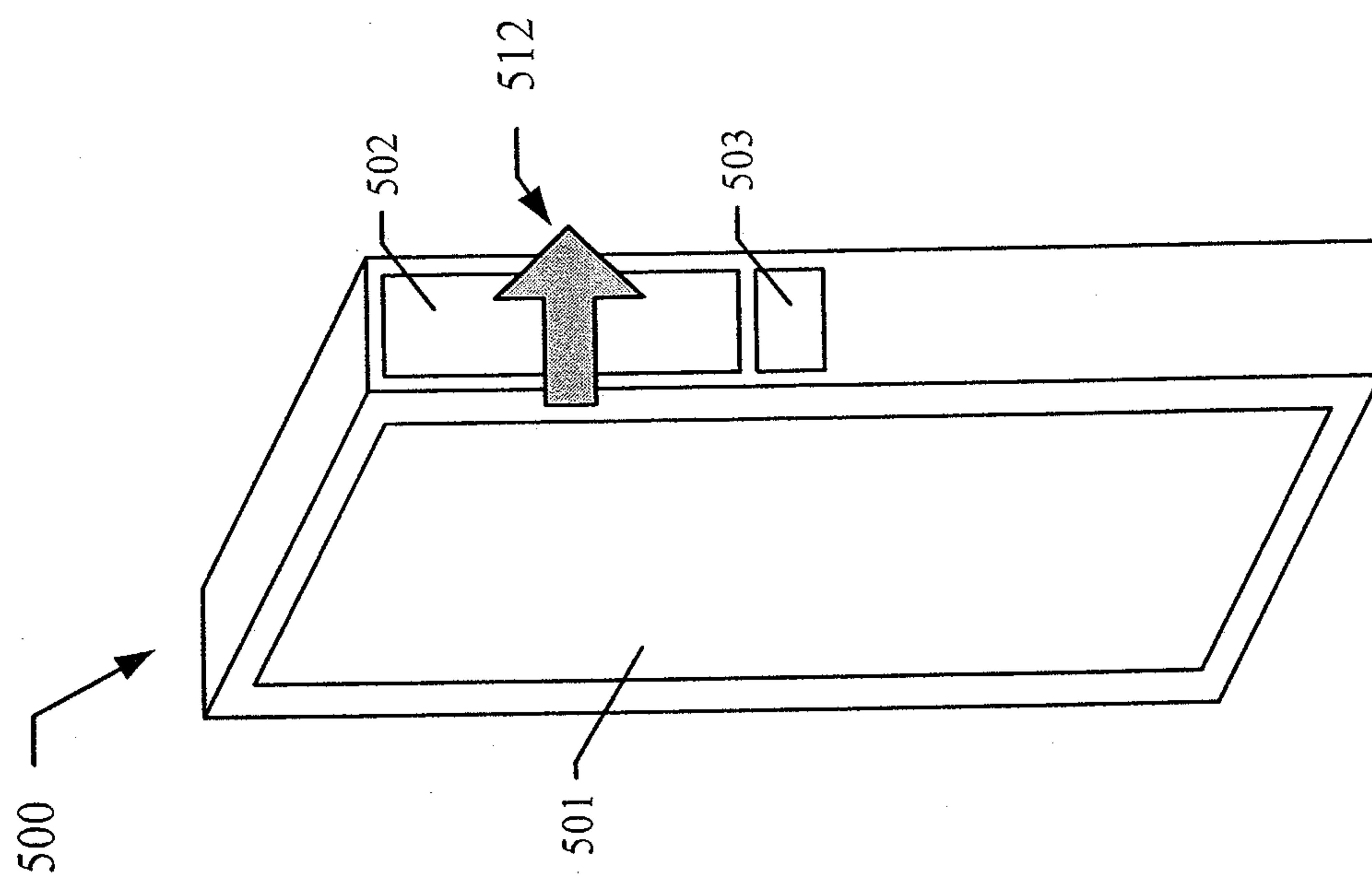


FIGURE 5B



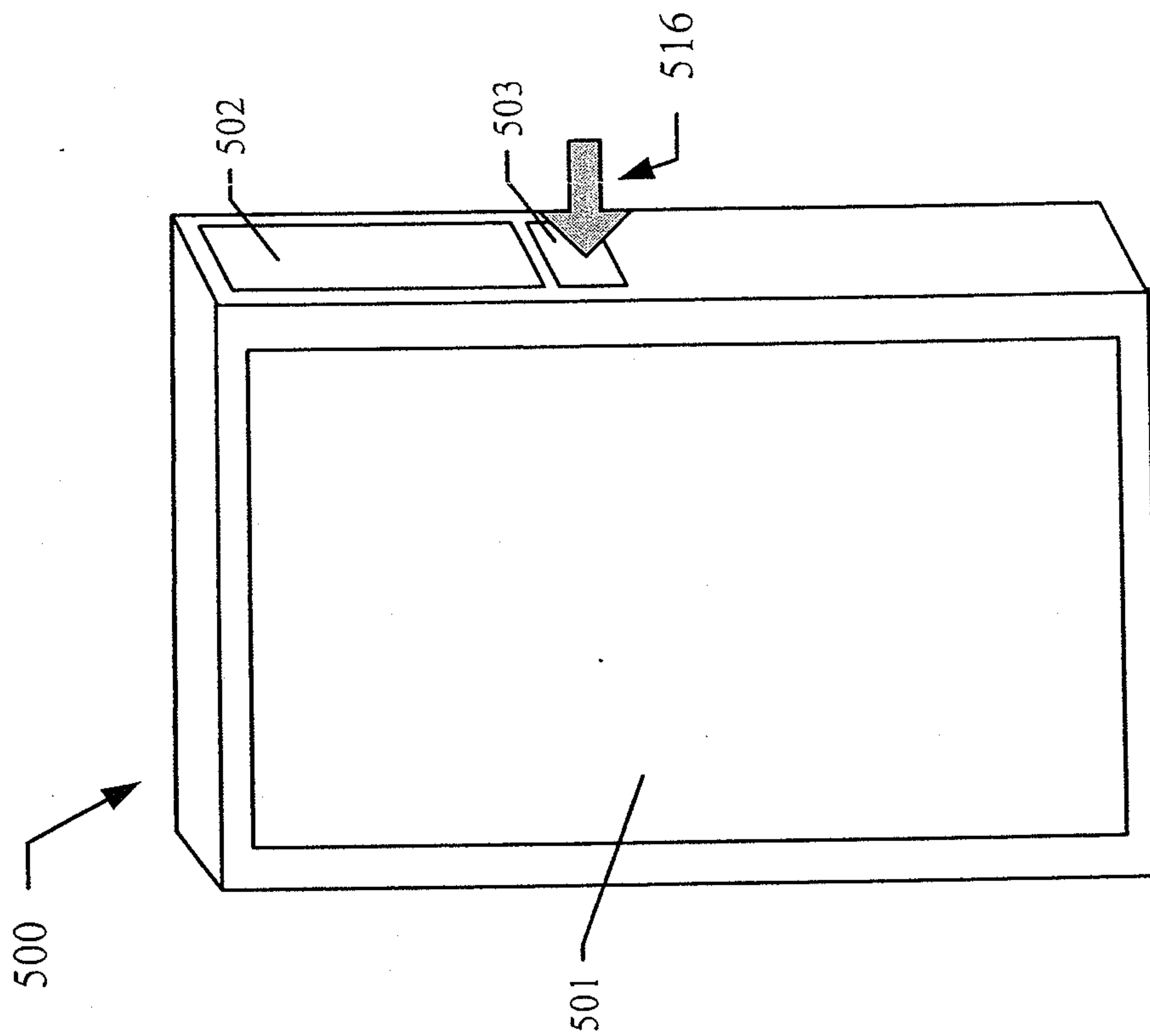


FIGURE 5F

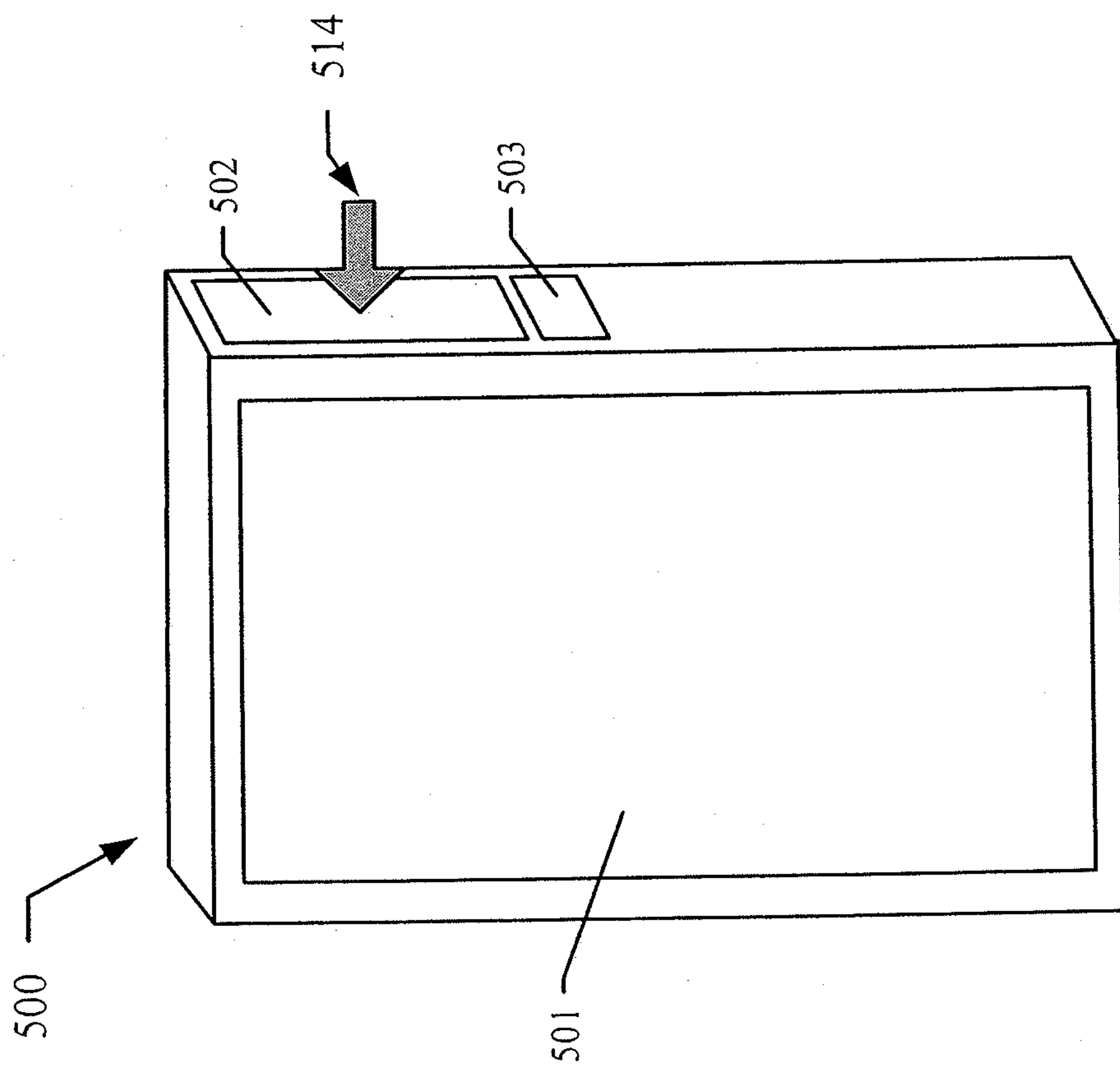


FIGURE 5E

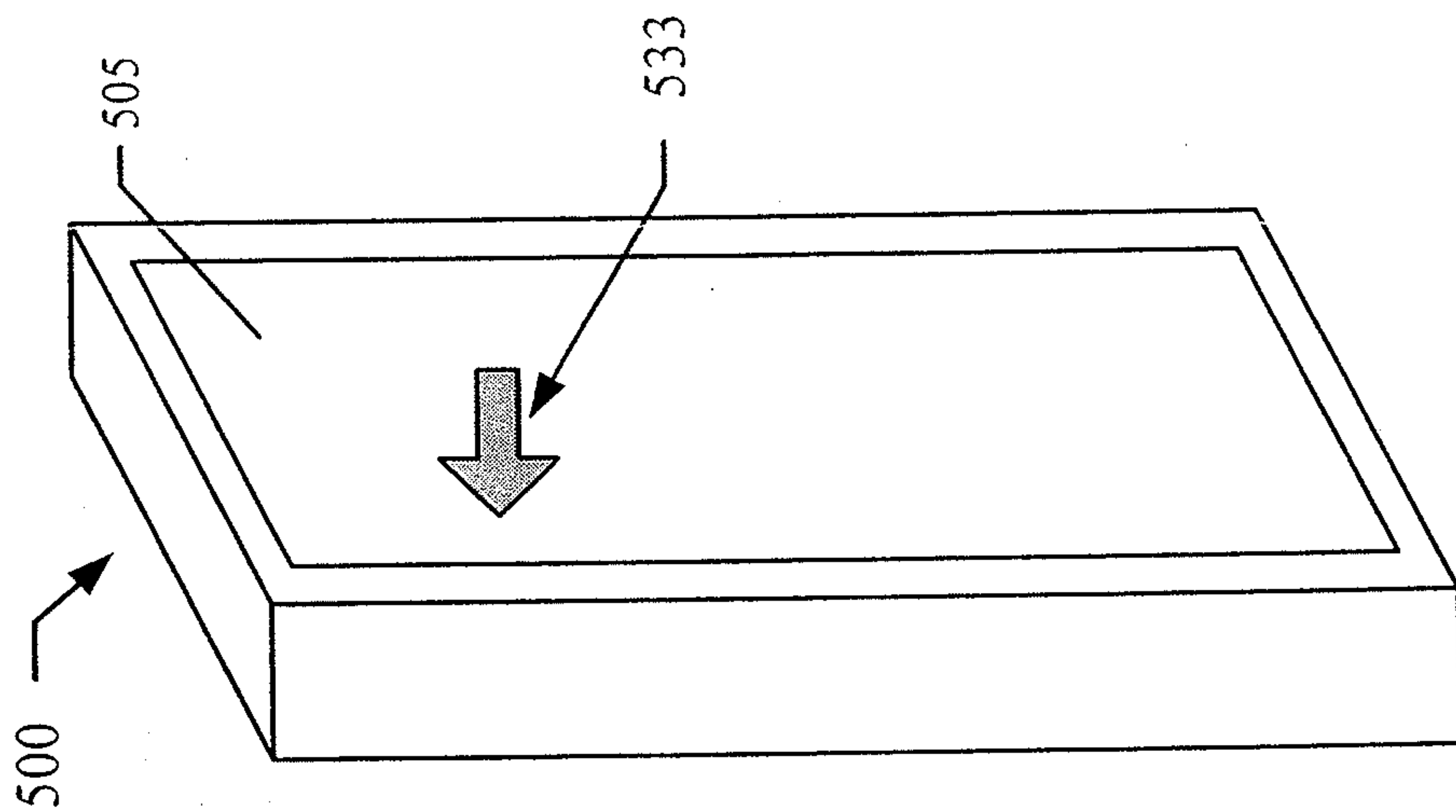


FIGURE 5I

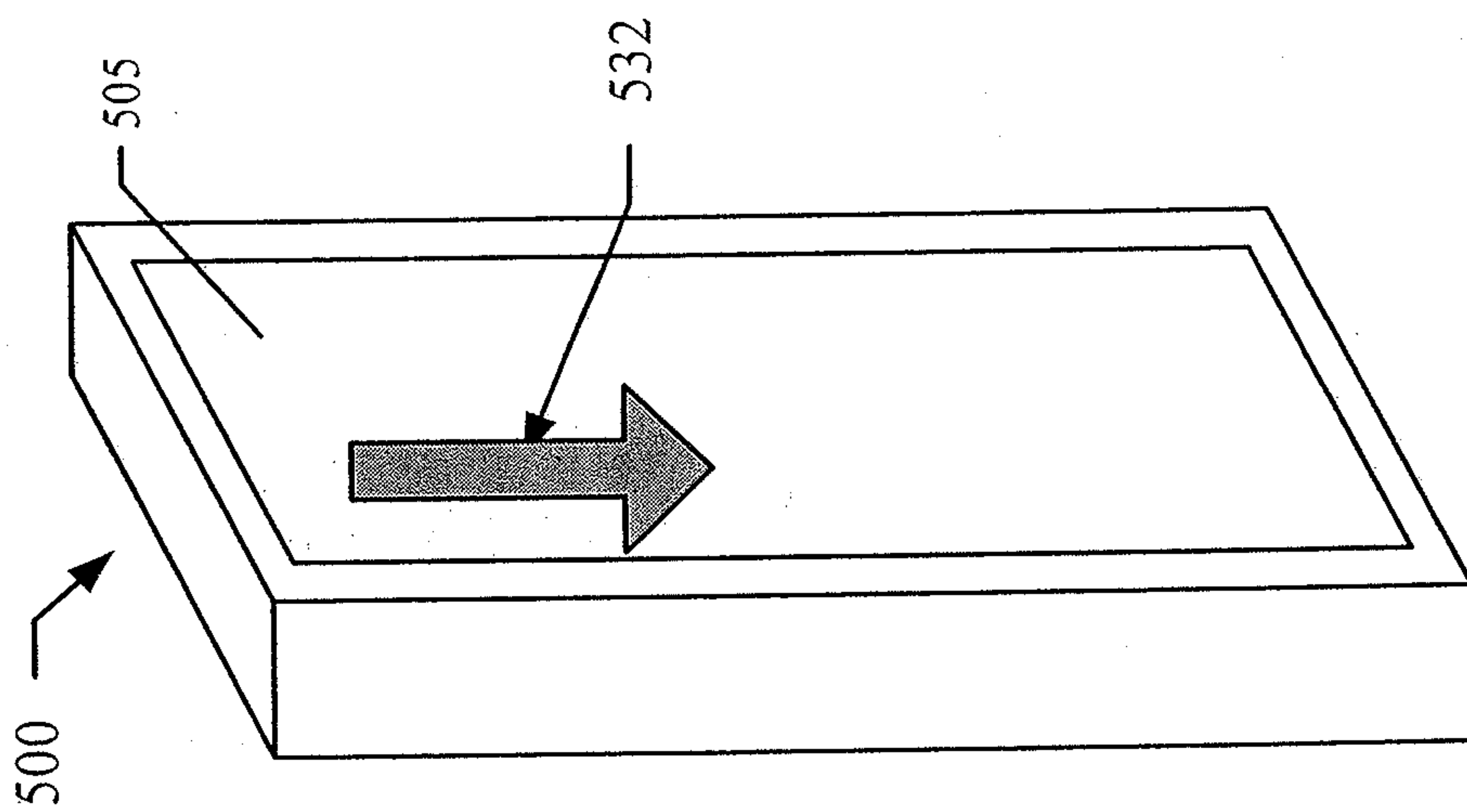


FIGURE 5H

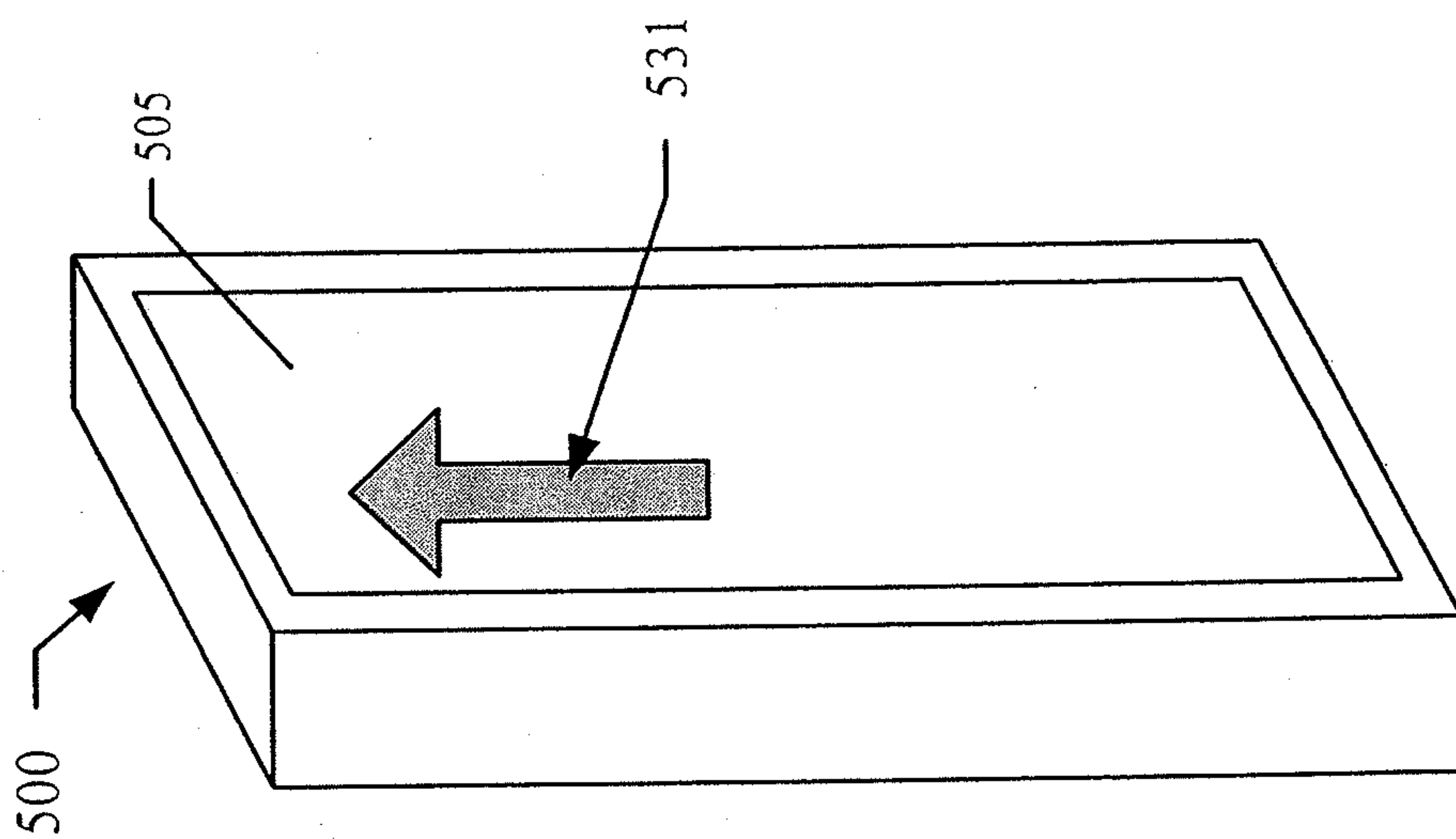


FIGURE 5G

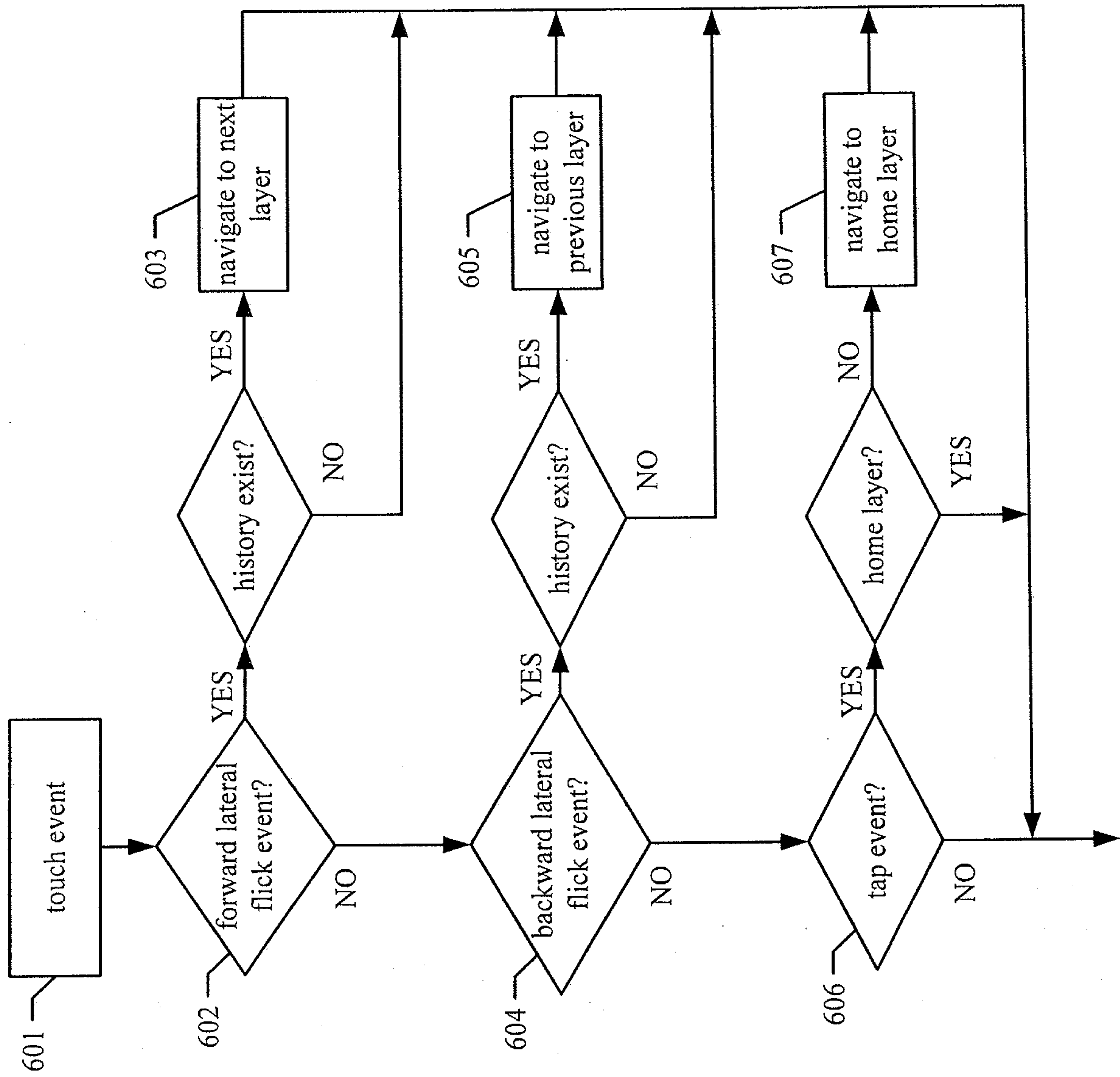


FIGURE 6

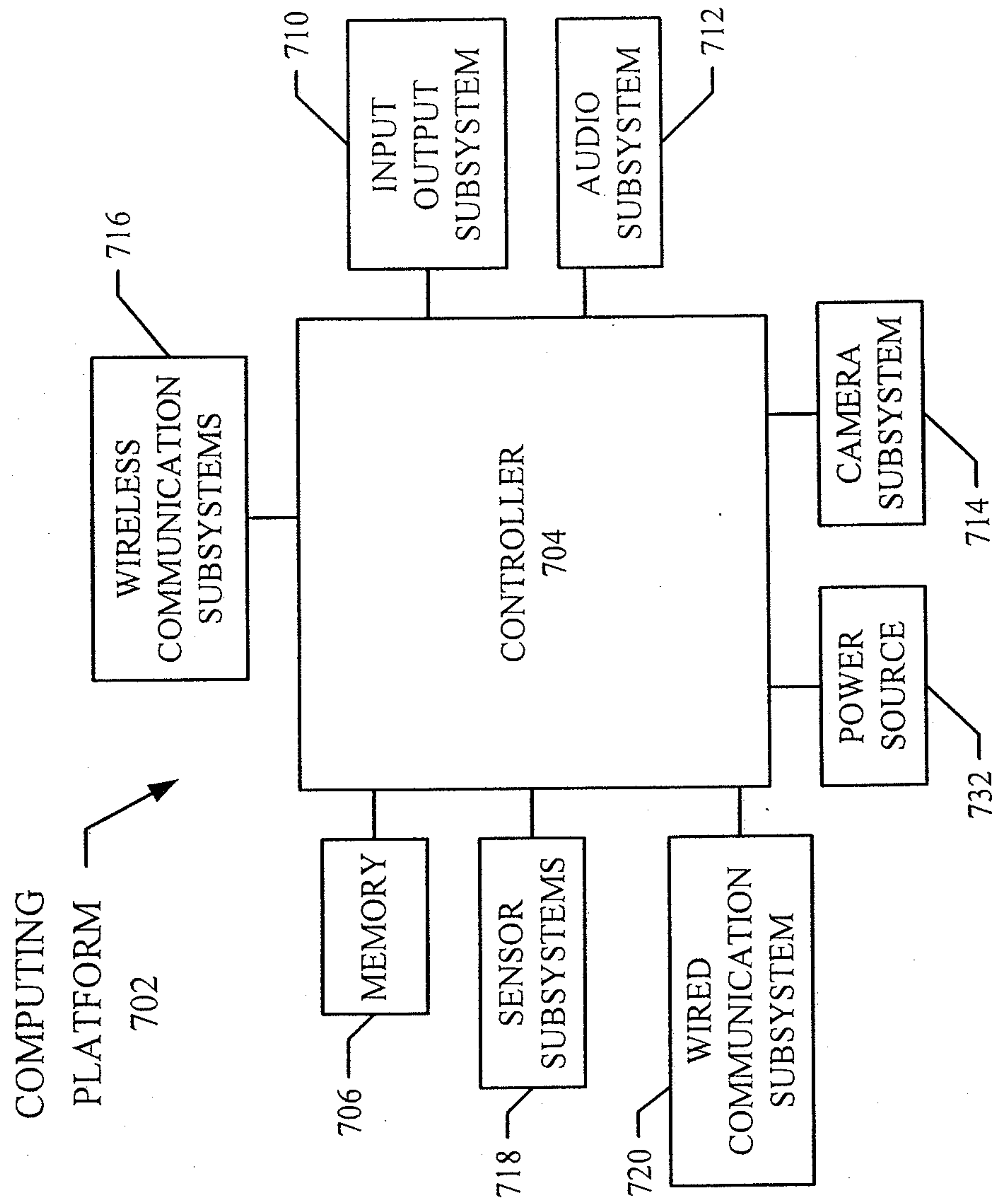


FIGURE 7

